



# Whistler Design Preview

## April 20 & 21



# **Internet Explorer Services and Platform Update**

**Dave Massy  
Internet Explorer  
Microsoft Corporation**

# Agenda

- **Introduction**
- **DHTML Enhancements**
- **Component Model**
- **HTML+Time**
- **Editing**
- **Call to Action**

# Introduction - Goals for 5.5

- **Build on success of Internet Explorer 5.0**
  - **Stable and Fast**
  - **Platform for Application UI - On and off the web**
- **Printing**
  - **Improved quality plus Print Preview**
- **Component Model**
  - **Extensibility hooks for custom HTML elements**
- **Editing**
  - **Extensibility hooks**
- **Continued focus on Performance**

# **DHTML Enhancements**

- **CSS support**
- **Vertical Text**
- **Popup**
- **Colored Scroll Bars**
- **Native Frame**
  - **Fast**
  - **Transparent**
  - **Play in the Z-Index**

# CSS support (1 of 2)

- **Borders, Margin and Padding on inline Elements**
- **Dotted and Dashed borders**
  - `border-style:dotted`
  - `border-style:dashed`
- **Zoom**
  - **Set the CSS zoom property on any element**
    - `zoom:150%` - 150% of original size
    - `Zoom:2` - two time original size

# CSS support (2 of 2)

- **First Letter and First Line CSS support**
  - Enables Newspaper like layout where the first letter and line are rendered in a different font size
- **Vertical Text**
  - Asian Markets
    - Japanese, Traditional Chinese



# Popup

- **Displaying a DIV always gets clipped by the client area**
  - Not good for context menu's and tooltips
- **New support for Popup object**
  - Create popup object
    - `pop = window.createPopup();`
  - show method - show the popup
  - Hide method -hide the popup
  - isOpen property - is it being shown?



# Colored Scroll bars

- Who wants chunky gray windows style scrollbars?
- Scrollbars can now have color CSS applied
  - scrollbar-base-color:red
  - scrollbar-shadow-color:yellow
  - scrollbar-3dlight-color:blue
  - scrollbar-highlight-color:green
  - scrollbar-darkshadow-color:pink
  - scrollbar-face-color:cyan
  - scrollbar-arrow-color:purple
- Create pages that have an integrated color scheme

# Native Frame

- **What's a native Frame?**
  - In Internet Explorer versions 3, 4 and 5.0 each frame was another instance of the browser
  - In Internet Explorer 5.5 each frame is within the same single instance of the browser
- **Why Change?**
  - **Performance**
    - Typical Frames pages 25-30% faster
    - Navigation of non frame pages faster

# IFRAMES

- **Transparency**

- IFRAME's document has transparent background

- `<BODY style="background-color:transparent">`

- **Z-Index support**

- IFRAME can have z-Index specified (previously it was always on top)

- `<IFRAME src='mycontent.htm' style='z-index:1'>`

- **Easier Frame access**

- `window.frameElement` - hosting frame element

# Unadorned dialogs

- **HTML Dialogs without the windows adornments**
  - Only from trusted host applications such as HTA or C++ hosts
  - Specify unadorned in the dialog options  
`showModalDialog('scc.htm','',  
    'unadorned:yes')`

# HTML+TIME / SMIL

## Boston

- Builds upon IE 5.0's HTML+TIME release
- Simplifies usage and integration of media content: images, audio, video, Windows Media, DVD
  - No need to use controls
- Enables synchronization of page elements with media
  - Apply timing to any tags
  - Declarative:
    - removes dependency on script
    - Interchange format for tools
- Support for animation of elements' properties, synchronized with media, including motion paths

# Enhancements to SMIL-Boston

- **Drives HTML+TIME into SMIL-Boston standard**
- **Any feature that is part of the standard, follows the standard**
- **MS extensions:**
  - **Runtime information**
    - **Properties and methods that facilitate authoring tool development, as well as applications that use script**
  - **Support for custom players**
    - **Enables new media formats to plug into the page's timing**

# Component Model

- **Element Behaviors**
  - Custom Elements as robust as standard HTML Elements
- **ViewLink**
  - Encapsulated DHTML UI for Behaviors
- **Additional Object Model**
  - Event firing, Defaults, editable etc.



# Attached Behaviors

- Internet Explorer 5.0 - Attached Behaviors
  - Separate content from script
    - Reusable engineering blocks
    - Manageable content
  - Behavior can be attached to *any* element
    - Using CSS behavior property or addBehavior method
  - Behavior can also be removed (sometimes unintentionally ☹)

# Element Behaviors (1 of 3)

- Evolution of Attached Behaviors
- Implement your own Custom Element
  - `<Namespace:TagName>`
- The Behavior is the Custom Element,  
The Custom Element is the Behavior
- Only a Custom Element can have an Element Behavior
- Only one Element Behavior per

# **Element Behaviors (2 of 3)**

- **Synchronous parsing**
  - **Behavior is available immediately**
- **Behavior cannot be detached from the Element**
- **Declared using Processing Instruction (PI) not CSS**

# Element Behaviors (3 of 3)

.htm file

```
<html xmlns:veg>
<?import
  namespace=veg
  implementation=tn
  1.htc>
<body>
<veg:turnip>This is
  my tag
</veg:turnip>
</body>
</html>
```

tn1.htc

```
<public:component
  tagName=turnip >
<!-- properties, methods,
  events and attach tags
  go in here -->
</public:component>
<script>
</script>
<body>
<!-- content can go here
  -->
```

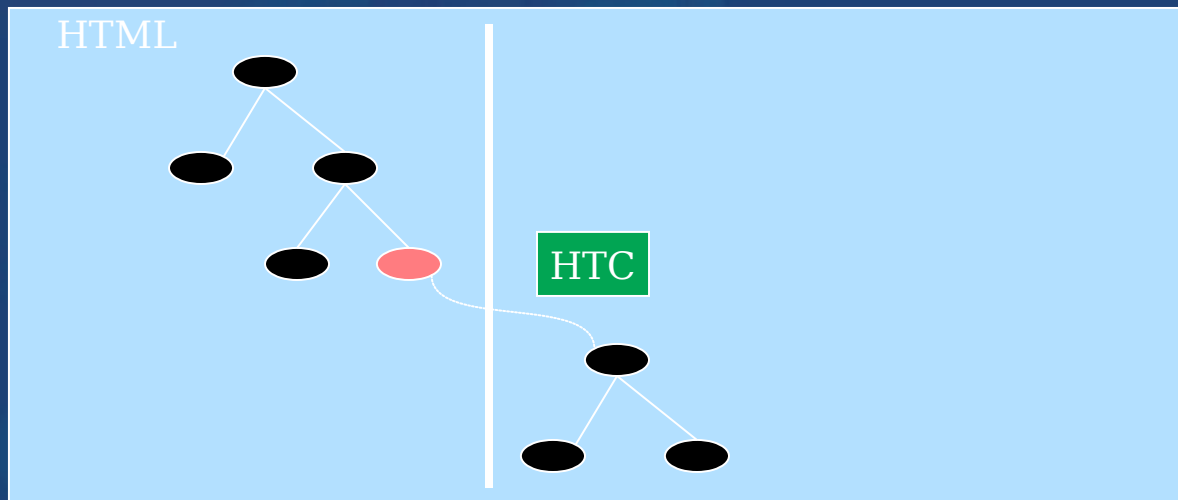
# ViewLink - (1 of 2)

- **The Problem**

- In IE5.0 a Behavior can display HTML by adding it to the document to which the behavior is attached. E.g. Calendar tag
- Pollution of the document structure
- Styles get applied inadvertently

# ViewLink - (2 of 2)

- **The Solution - ViewLink your content**
  - Create the HTML content in a separate document fragment
  - Use that fragment as the view for the Element



# Declarative ViewLinking

- HTML content inside the HTC

```
<PUBLIC:COMPONENT tagName='mytag'
>
  <PUBLIC:DEFAULTS
    viewLinkContent='true' />
</PUBLIC:COMPONENT>
<BODY>
  <B>This content will show up as the
    content of Mytag using ViewLink</B>
</BODY>
```



# Procedural ViewLinking

- Create the document fragment
- Set the ViewLink to that fragment

```
<PUBLIC:COMPONENT tagName='mytag' >  
<PUBLIC:ATTACH event=oncontentready  
  onevent="init();" />
```

```
</PUBLIC:COMPONENT>
```

```
<SCRIPT>
```

```
function init() {
```

```
  var nc = document.createElement("B");
```

```
  var text =
```

```
    document.createTextNode("Content of my  
    ViewLink");
```

```
  nc.insertBefore(text, null);
```

```
  defaults.viewLink = nc.document;
```

```
}
```

```
</SCRIPT>
```

# Defaults

- Each tag has some default properties
  - A `<b>` tag has a default style of `font-Weight:bold`
  - A `<div>` tag has a default style of `display:block`
- So that means
  - `<b style="display:block; font-Weight:normal">`  
== `<div>`
  - `<div style="display:inline; font-Weight:bold">`  
== `<b>`

# Default Properties

- ***Element Behaviors*** are new tag implementations, they require the ability to declare their *default properties*.
- **Which properties can have defaults?**
  - **style** - all css properties
  - **contentEditable** - an element's content is running in editmode
  - **containsSelection** - when text selection is started, can it move outside of element?
  - **isMultiLine** - can an element's content wordwrap?
  - **tabStop** - should the element be in tab order?

# Declarative Default Properties

Building an <ie:bi> HTC

```
<public:component tagName=bi >
```

```
<public:defaults style="font-  
    Weight:bold;font-Style:italic" >
```

```
</public:component>
```

# Programmatic Default Properties

- **Defaults object**
  - **HTC:** Available in the behavior script namespace
  - Has a corresponding property for each declarative default property
  - **Examples:**
    - `defaults.style.color = "blue";`
    - `defaults.contentEditable = "true";`

# Editing

- **Build Rich HTML data entry components**
- **Any Element can be switched in to Edit mode**
  - **contentEditable property**
  - **Inherited by child elements**
- **An Element's edit mode can be queried**  
**isContentEditable property**

# **Additional Component Model capabilities:**

- **Literal Content**
  - **Allow an Element to have content not parsed and built into the document.**
  - **E.g. <XML> and <SCRIPT> elements**
  - **Advanced uses only as the element's content will have limited access from script**



# **Other New stuff but not for scriptors**

- **New Rendering and Layout interfaces for Binary Behaviors**
- **Filters and Transitions use DirectX Transforms**
- **Printing improvements**
- **Print Preview**

# Editing for hosts:

- **Host mshtml.dll for editing**
  - **Multiple selection**
  - **Custom Glyphs**
  - **Pluggable editing DLLs**
    - **Override default editing experience**
    - **Add value**
    - **E.g.**
      - **WYSIWYG Table Editing**
      - **Background spell check**

# Resources

- **MSDN**  
**Internet Explorer Developer Center**  
**<http://msdn.microsoft.com/ie>**
- **Internet Explorer 5.5 Beta download**
- **Overviews and Articles**
- **Full Reference Documentation**
- **Samples**

# Call to Action

- **Use Internet Explorer 5.5**
  - **Build your own robust Custom Elements**
    - **Element Behaviors**
    - **ViewLink**
  - **Visual effects**
    - **Colored scrollbars**
    - **Transparent z-indexable Iframes**
    - **HTML+Time, Etc. etc. etc.**
- **Give us your feedback**
  - **On the product**
    - **Compatibility issues**
    - **New Features**
  - **On the documentation**

**Where do you want to go today?®**

**Microsoft®**